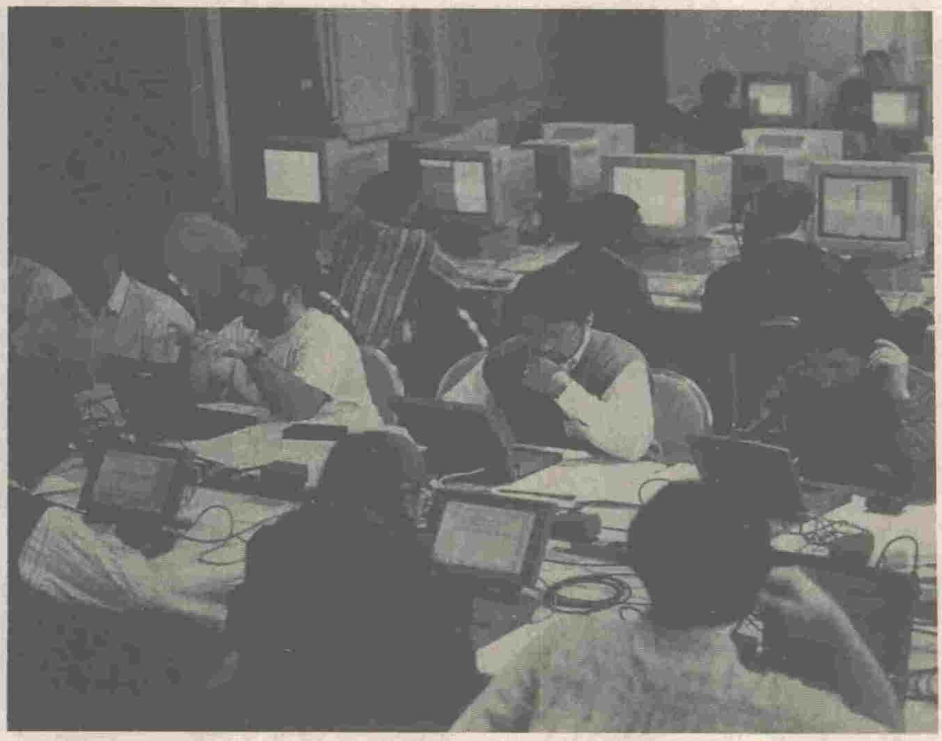
ปัญญา เปรมปรีดิ์

หลักการเขียนโปรแกรมเกมคอมพิวเตอร์

หลักเกณฑ์และขั้นตอนที่จะพัฒนาตนเองไปสู่การเป็นผู้เขียนโปรแกรมเกม แต่จะให้เกิดทักษะหรือความชำนาญนั้นต้องใช้เวลา



มีคนโทรไปถึงผมและถามถึงเรื่องวิธีเขียนโปรแกรมเกมคอมพิวเตอร์อยู่หลายท่าน บางท่านก็สนใจเฉพาะเจาะจงลงไปเป็นเรื่องของหมากรุกไทย ซึ่งผมก็ได้พยายามอธิบายไปเท่าที่พอจะทำได้ทางโทรศัพท์ และผมมักจะบอกพวกเขาให้รออ่านเรื่องเหล่านี้ ผมมีความตั้งใจอยากจะเขียนตำราหรือบทความเกี่ยวกับเรื่องนี้มานานแล้ว บางส่วนผมก็ได้เขียนลงวารสารไปบ้าง แต่ฉบับที่สมบูรณ์จริง ๆ นั้นยังไม่มี มาคราวนี้กองบรรณาธิการของวารสารไมโครฯ ก็เกิดโทรมาบอกว่า ฉบับหน้านี้เขาจะพูดถึงเรื่องเทคนิคการเขียนโปรแกรม และเขาอยากให้ผมพูดอะไรที่โยงมาถึงเกมคอมพิวเตอร์สักหน่อยหนึ่ง ผมก็เลยพยายามทำดู แต่พอเขียนไปได้สัก 20 หน้าก็พบว่า การที่จะเขียนเรื่องอย่างนี้ให้สมบูรณ์จริง ๆ นั้นเป็นไปไม่ได้ มันจะกลายเป็นบทความที่ยาวเกินไป ผมก็เลยต้องมาเริ่มต้นกันใหม่ ผมคงจะพูดได้แค่หลักการ คงจะลงในรายละเอียดของโปรแกรมหรือตัวอย่างไม่ได้ และผมก็คงต้องให้สัญญาเอาไว้อีกว่า ผมจะพยายามเขียนเป็นตำราชนิดร่ายยาวตั้งแต่ซื้อเครื่องไปจนถึงขั้นเขียนโปรแกรมเกมได้ด้วยตนเองให้ท่านอ่านในวันหลัง

โอเค เรามาดูเรื่องหลักการหรือแนวทางที่จะเขียนเกมคอมพิวเตอร์กัน เราคงจะต้องเริ่มต้นที่เรื่องของ**การเลือกภาษาเสียก่อน** ในปัจจุบันนี้เครื่องเกือบทั้งหมดวิ่งภายในระบบปฏิบัติการวินโดวส์ แต่บางที (20-30 เปอร์เซ็นต์ของเวลา) เราก็ต้องออกไปใช้ดอส ทั้งนี้เพราะมีแอพพลิเคชันบางอย่างที่เป็นของเก่าและพัฒนาภายใต้ดอส แต่แนวโน้มนั้นจะเลิกใช้โปรแกรมเก่า ๆ ไป และหันมาซื้อชุดใหม่ที่วิ่งภายใต้วินโดวส์มากขึ้น ผมอยากจะย้ำคำว่า

"ซื้อ" ทั้งนี้เพราะโอกาสที่ท่านจะพัฒนาโปรแกรมภายใต้วินโดวส์ด้วยตนเองนั้นมีน้อยมาก ปัญหาของการพัฒนาโปรแกรมบนวินโดวส์นั้นมีหลายอย่าง ปัญหาข้อแรกคือโปรแกรมบนวินโดวส์นั้นแยกเป็น 2 ส่วน ส่วนหนึ่งเป็นการสร้างภาพบนวินโดวส์และโต้ตอบกับผู้ใช้ ส่วนนี้จะทำบนจอภาพเลยทีเดียว (Visual Programming) การเขียนโปรแกรมส่วนนี้ท่านต้องเรียนรู้การใช้คำสั่งบนเมนูบาร์ คำสั่งในไดอะล็อกบ็อกซ์ และคอนโทรลต่าง ๆ ที่เป็นไปตามมาตรฐานการทำงานภายใต้วินโดวส์ มันเป็นส่วนที่ต้องใช้เวลาเรียนรู้สูง และท่านจะหาคู่มือหรือตำราเพื่อตอบคำถามต่าง ๆ ให้ตรงจุดได้ยากมาก ตำราเกี่ยวกับวินโดวส์และการเขียนโปรแกรมบนวินโดวส์นั้นมีเยอะแยะ แต่อ่านไม่ค่อยรู้เรื่อง และจะไม่มีอะไรที่ตรงกับปัญหาที่ท่านพบอยู่ สำหรับโปรแกรมส่วนที่ 2 นั้นเป็นโปรแกรมที่ถูกเรียกโดยอีเวนต์ (Event) ต่าง ๆ เช่น เมื่อผู้ใช้คลิกที่ไอคอนบนจอภาพ หรือไทม์เมอร์มันถึงเวลาที่เรากำหนดไว้ โปรแกรมพวกนี้ต้องเขียนด้วยภาษา C หรือ Pascal หรือ C++ และจะต้องเป็นเวอร์ชันใหม่ ๆ เท่านั้น อย่าง Pascal ใน Delphi ก็จะต้องเป็นเวอร์ชัน 7.0 ขึ้นไป ภาษาระดับนี้เข้าใจยาก เขียนยาก โดยสรุปแล้วปัญหาแรกของการเขียนโปรแกรมบนวินโดวส์ก็คือ มันเขียนยาก มันเข้าใจยาก มันใช้เวลาแก้ปัญหายาวนานมาก ผมลองกับ Delphi มาหลายครั้งแล้ว ผมต้องเก็บพักเอาไว้ก่อนทุกครั้ง

ปัญหาของการเขียนโปรแกรมบนวินโดวส์ประเภทที่ 2 ก็คือ โปรแกรมย่อยที่ถูกเรียกโดยอีเวนต์ (Event Driven Program) เหล่านี้อาจถูกเรียกขึ้นมาทำงานขนานกันได้ ทั้งนี้มันขึ้นอยู่กับอีเวนต์หรือเหตุการณ์ที่เปลี่ยนแปลงไปต่าง ๆ และบางทีมันก็เรียกตัวอื่นขึ้นมาทำงานขนานเพิ่มขึ้นไปอีกได้ สิ่งเหล่านี้ทำให้เกิดโพรเซสหลาย ๆ โพรเซสขนานกัน เราจะไม่มีหนทางบังคับว่าอะไรจะจบก่อนอะไร การวางขั้นตอนการทำงานจึงยุ่งยากและสับสนได้ง่าย มันไม่เป็นไปตามแนวคิดเดิม ๆ ที่เราเรียกว่า ซีเควนเชียลแมชีน (Sequential Machine) ซีเควนเชียลแมชีน คือเครื่องมือหรือโปรแกรมที่ทำงานทีละคำสั่งตามลำดับที่เราวางไว้ คนที่จะเริ่มหัดเขียนโปรแกรมนั้นควรเริ่มกับซีเควนเชียลแมชีนเสียก่อน ท่านจึงต้องเขียนได้ดี คิดวางแผนเป็นขั้นตอนได้ดี และทำให้เกิดโปรแกรมตามความคิดนั้นได้จริงเสียก่อน แล้วจึงจะไปหัดคิดและเขียนโปรแกรมที่มีโปรแกรมย่อยหลายตัววิ่งขนานกัน

ปัญหาของวินโดวส์ในข้อที่ 3 ก็คือ โปรแกรมย่อยที่เราเขียนด้วยภาษา C หรือ Pascal หรือ C++ นั้น มันไม่สามารถไปวาดรูปหรือทำอะไร ๆ บนจอภาพได้โดยตรง มันต้องส่งเป็นคำขอไปยังวินโดวส์ นี่อาจเป็นข้อดีในแง่ของการควบคุม คือวินโดวส์จะควบคุมจุดทุกจุดบนจอภาพหมด ใครจะมาทำอะไรเองไม่ได้ วินโดวส์เท่านั้นที่จะแก้ไขภาพได้ แต่เมื่อมองในแง่ของการเขียนโปรแกรมแล้ว มันเข้าใจยาก มันสร้างคำสั่งหรือคำขอไปยังวินโดวส์ได้ยาก และมันต้องมีการทดสอบผลการดำเนินงานของวินโดวส์ เจ้าโปรแกรมย่อยแต่ละตัวนี้จริง ๆ แล้วเป็นซีเควนเชียลแมชีน คือตัวมันต้องทำงานไปทีละคำสั่ง การติดต่อกับวินโดวส์นั้นทำแล้วต้องเช็กผลเสียก่อนที่จะดำเนินในขั้นตอนถัดไป มันจึงสร้างความยุ่งยากและสับสนได้มาก

ครับ จากปัญหา 3 ข้อที่กล่าวมานี้ ผู้ที่คิดจะลองเขียนโปรแกรมเล่นเกมจึงต้องเริ่มจากโปรแกรมบนดอสเสียก่อน ภาษาที่ใช้กับดอสนั้นมีให้เลือกมากพอสมควร เช่น Basic, Pascal, C แต่ละภาษามีผู้ผลิตหลายราย เช่น ไอบีเอ็ม ไมโครซอฟท์ บอร์แลนด์ ซันไมโครซิสเต็ม แอปเปิ้ล และแต่ละผู้ผลิตนี้ก็ผลิตออกมาหลายเวอร์ชัน เราจึงต้องมีหลักเกณฑ์ในการเลือกภาษา หลักเกณฑ์ที่สำคัญ ๆ มีดังนี้

1. มันต้องวาดรูปเป็นภาพสีบนจอภาพได้อย่างรวดเร็ว สิ่งที่เราต้องการมี 7 อย่าง ดังนี้

1.1 เปลี่ยนสีของพิกเซลทีละพิกเซลได้

1.2 เช็กว่าสีของพิกเซลอันใดอันหนึ่งเป็นสีอะไรได้

1.3 ลากเส้นตรงจากจุดหนึ่งไปยังอีกจุดหนึ่งได้

1.4 เก็บภาพบนพื้นที่สี่เหลี่ยมผืนผ้าที่วางตามแนวนอนหรือแนวตั้งเข้าไว้ในเมมโมรีได้

1.5 นำภาพที่เก็บไว้ในเมมโมรีไปแสดงออกที่บริเวณอื่นบนจอภาพได้

1.6 นำภาพที่เก็บไว้ในเมมโมรีออกไปบันทึกเป็นเรคอร์ดบนดิสก์ได้

1.7 อ่านข้อมูลภาพที่เก็บไว้ในดิสก์กลับขึ้นมาใส่ในเมมโมรีได้

การที่เราต้องการความสามารถ 7 อย่างนี้ก็เพราะเกมคอมพิวเตอร์นั้นส่วนใหญ่ต้องสร้างภาพเป็นภาพสี และต้องเปลี่ยนแปลงภาพเหล่านี้ได้อย่างรวดเร็ว ทั้งนี้เพื่อให้เกิดเป็นภาพเคลื่อนที่ได้ และเกมคอมพิวเตอร์นั้นบางทีเราต้องเตรียมภาพเอาไว้โดยโปรแกรมอื่น ๆ เสียก่อน แล้วจึงมาเรียกใช้โดยโปรแกรมที่จะเล่นเกมกับเรา มันจึงต้องสามารถบันทึกภาพลงดิสก์ และเรียกกลับมาทีหลังได้

2. มันต้องมีความสามารถในการสร้างตัวเลขสุ่ม (Random Number) ได้เร็วและกระจายดี ตัวเลขสุ่มนั้นมีไว้เพื่อการทำให้เกมเลือกหนทางปฏิบัติที่แปรเปลี่ยนไปได้โดยที่เราไม่สามารถจะคาดเดาได้ เกมจะเริ่มใหม่โดยไม่ซ้ำแบบเดิม

3. มันต้องมีความสามารถในการติดต่อขอวัน (System Date) และขอเวลา (System Time) จากดอสได้ เกมนั้นต้องมีการจัดจังหวะของการแสดงภาพ แสดงเสียง มีการให้คะแนนตามความเร็วในการทำงานให้สำเร็จ เกมจึงต้องเช็กเวลากับดอสอยู่เกือบตลอดเวลา คำสั่ง DELAY ซึ่งจะช่วยหน่วงเวลาของการทำงานของโปรแกรมนั้นอาจไม่เหมาะในหลาย ๆ กรณี ทั้งนี้เพราะโปรแกรมจะหยุดที่คำสั่งนี้โดยไม่ไปทำอะไรอย่างอื่น เกมนั้นจะต้องทำโน่นทำนี่อยู่ตลอดเวลา อย่างน้อยก็ต้องรับตัวอักษรจากแป้นคีย์บอร์ดมาดำเนินการไปก่อน ขณะที่ทำไปนั้นก็ต้องเช็กเวลากับดอสเพื่อเปลี่ยนภาพ ฯลฯ

4. มันต้องมีความสามารถในการดึงข้อมูลจากคีย์บอร์ดบัฟเฟอร์โดยตรง เกมส่วนใหญ่จะกดเลือกหรือสั่งด้วยการกดแป้นเพียงครั้งเดียว เราจะไม่ใช้คำสั่ง READ ทั้งนี้เพราะคำสั่ง READ ต้องกดข้อมูลและแป้นเอ็นเตอร์ (Enter, ↵) ซึ่งจะเสียเวลา การกดแป้นคีย์บอร์ดนี้จริง ๆ แล้วเราสามารถกดไว้ล่วงหน้าได้ถึง 12 ตัวอักษร รหัสตัวอักษรที่เรากดเข้าไปนี้จะเข้าไปเก็บค้างอยู่ในเมมโมรี ส่วนที่เราเรียกว่า คีย์บอร์ดบัฟเฟอร์ การเรียกขอจากโปรแกรมแต่ละครั้งจะได้ไป 1 ตัวอักษร (แบบ first in-first-out) และบัฟเฟอร์ก็จะเลื่อนไปโดยอัตโนมัติ

5. มันต้องมีความสามารถในการทำเสียงออกที่ลำโพงเล็กภายในเครื่องได้ หรือถ้าออกที่ซาวด์การ์ดได้ก็ยิ่งดี นี่เป็นเพราะเกมนั้นต้องการกระตุ้นอารมณ์ของผู้เล่นด้วย แสง-สี-เสียง-ความเร็ว-และความคิด

6. มันต้องง่ายต่อการนำโปรแกรมเข้าแก้ไขโปรแกรม และทดสอบโปรแกรม การทำเกมนั้นเป็นความคิดชั่วแล่น คือคิดอะไรต้องรีบทำ ถ้ามาติดขัดในเรื่องการเขียนและการทดสอบโปรแกรม มันก็จะลืมไปได้ง่าย ๆ

7. มันต้องเป็นคอมไพเลอร์ที่มีประสิทธิภาพสูง เกมส่วนใหญ่ต้องทำงานเร็ว ถ้ามันทำงานช้า ยืดยาด ผู้เล่นจะไม่สนุก จะมียกเว้นก็แต่เรื่องที่ต้องใช้ความคิดจริง ๆ จัง ๆ เท่านั้น ส่วนใหญ่แล้ว ถ้าเป็นเกมแบบแข่งกันระหว่างคนกับคอมพิวเตอร์เราจะพบว่าประสิทธิภาพเป็นเรื่องสำคัญมาก เช่น เกมหมากรุก เครื่องมักจะใช้เวลาคำนวณนานมาก ถ้าโปรแกรมทำงานเร็วขึ้นก็จะคิดได้ลึกยิ่งขึ้น ซึ่งจะเกิดความท้าทายสูงขึ้น

ครับ ที่กล่าวมานั้นเป็นหลักการที่สำคัญ และผมพบว่า Pascal เวอร์ชัน 3.0 ของบริษัทบอร์แลนด์นั้นเหมาะสมที่สุด ผมสามารถใช้มันพัฒนาเกมได้เกือบทุกรูปแบบ มันมีข้อจำกัดอยู่เล็กน้อย คือมันจำกัดอยู่ในขอบเขตเพียง 64K byte เท่านั้น แต่มันก็พอจะใช้งานได้ ข้อเด่นของมันคือพัฒนาได้เร็ว และมีประสิทธิภาพสูง

โอเค หลังจากเลือกภาษาได้แล้วเราก็**คงจะต้องสร้างความชำนาญในการใช้อีดิตเตอร์ (Editer, ตัวรับและแก้โปรแกรม) และคอมไพเลอร์ (Compiler, ตัวแปลภาษาเป็นภาษาเครื่อง) และคำสั่งต่าง ๆ ในภาษานี้** ซึ่งเรื่องนี้ผมคงไม่สามารถจะนำมาบรรจุในบทความอันนี้ได้ การสร้างความชำนาญตรงนี้ส่วนใหญ่จะใช้เวลาตั้งแต่ 1 ปีไปจนถึง 3 ปี นี่ดูจากประสบการณ์ของพวกที่เข้ามายึดอาชีพเป็นโปรแกรมเมอร์ตามศูนย์คอมพิวเตอร์ต่าง ๆ แต่ถ้าเรารู้จักเลือกใช้เพียงบางส่วนของภาษา Pascal เวอร์ชัน 3.0 ของบอร์แลนด์นี้ เราก็สามารถจะเรียนรู้ได้ในเวลา 1 ถึง 3 สัปดาห์ เรื่องนี้คงจะต้องรออ่านจากตำราที่ผมจะเขียนในอนาคต

ทีนี้ก็มาถึงขั้นตอนที่ 3 คือ**เราจะต้องเลือกว่าจะเขียนเกมประเภทไหน** เกมนั้นมี 4-5 ประเภทใหญ่ดังนี้

1. อาร์เคดเกม เกมพวกนี้เป็นประเภทที่แสดงภาพรวดเร็ว ผู้เล่นจะต้องกดแป้นเพื่อทำให้เกิดเหตุการณ์ที่ถูกต้อง เช่น ยิงให้ถูกเป้า บังคับให้ตัวเขมือบวิ่งไล่จับตัวประหลาดให้ทันและจับกิน บังคับรถตักดินให้วิ่งหนีตัวเขมือบ... เกมพวกนี้ต้องให้ภาพที่สวยงาม เคลื่อนไหวได้รวดเร็ว และแปรเปลี่ยนโดยคาดเดาได้ยาก (แต่พอจะหาแนวโน้มได้เล็กน้อย

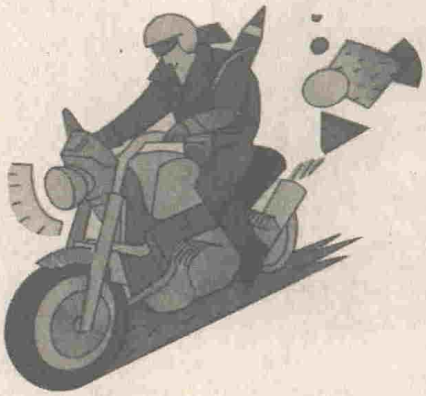
2. บอร์ดเกม นี่คือเกมประเภทเล่นบนกระดานที่เป็นตาราง และมีผู้เล่น 2 ด้าน ด้านหนึ่งคือผู้เล่นเกม อีกด้านหนึ่งคือคอมพิวเตอร์ บนกระดานจะมีตัวหมากชนิดต่าง ๆ วางอยู่ ตัวหมากเหล่านี้จะเคลื่อนที่ไปได้ตามกฎเกณฑ์ต่าง ๆ ที่กำหนดไว้ ผู้เล่นจะสลับกันเดิน หรือคือเคลื่อนย้ายตัวหมากไปครั้งละ 1 ตัว 1 ทิศทาง ตัวหมากของฝ่ายผู้เดินจะกินตัวหมากของฝ่ายตรงข้ามได้ตามกฎเกณฑ์ที่กำหนดไว้ เกมจะจบลงเมื่อกินตัวหมากที่สำคัญ หรือกินตัวหมากทั้งหมดของฝ่ายตรงข้ามได้ เกมชนิดนี้ผู้เล่นไม่ต้องใช้ความเร็วของสายตาและนิ้วมือ แต่ต้องใช้ความคิด ส่วนด้านคอมพิวเตอร์นั้นจะต้องคิดหาหนทางเดินที่ดีที่สุด ซึ่งมักจะต้องทำการคำนวณมากมาย ผู้เขียนต้องรู้จักวางขั้นตอนการคำนวณให้มีประสิทธิภาพ ใช้การคำนวณเป็นจำนวนครั้งต่ำ ๆ

3. ซิมูเลชันเกม นี่เป็นเกมที่วาดภาพบนจอภาพให้เหมือนกับที่เราจะมองเห็นได้จากการขับขี่หรือควบคุมเครื่องมือหรือเครื่องจักร ตัวอย่างเกมประเภทนี้คือ การขับเครื่องบิน การขับรถแข่ง การขับยานอวกาศ ภาพบนจอนั้นส่วนใหญ่เปลี่ยนแปลงไม่เร็วนัก แต่จะต้องให้ภาพที่ดูเหมือนจริง ภาพจะต้องเปลี่ยนไปตามการเปลี่ยนแปลงตำแหน่งของตัวควบคุม เช่น คันบังคับของเครื่องบิน พวงมาลัยของรถแข่ง คันบังคับของยานอวกาศ ความท้าทายของเกมพวกนี้คือความละเอียดของการบังคับตัวควบคุม มันจะต้องทำให้ดูยากเท่า ๆ กับของจริง

4. โรลเพลอิงเกม (RPG, Role Playing Game) นี่เป็นเกมที่ให้ผู้เล่นสามารถเลือกเป็นตัวละครตัวใดตัวหนึ่งในนิยายที่สร้างขึ้น ภาพบนจอจะเป็นสิ่งที่ผู้เล่นมองเห็น แต่ส่วนใหญ่

ก็เป็นลักษณะสัญลักษณ์มากกว่า คือจะเป็นภาพที่มีสิ่งของ หรือสัตว์ป่า สัตว์ประหลาด หรือเครื่องมือวางอยู่ มันจะไม่ค่อยเหมือนจริงเท่าใดนัก ภาพพวกนี้ส่วนใหญ่ไม่มีการเคลื่อนไหว บางภาพอาจมีข้อความที่จะใช้เป็นแนวทางในการแก้ปัญหา สิ่งที่ผู้เล่นทำได้มี 2-3 อย่าง เช่นเลือก (สั่งให้) หยิบเครื่องมือ หรือกดปุ่มหรือผลักชิ้นส่วนในภาพ หรือเลือก (สั่งให้) เดินผ่านช่องทางหรือประตูหน้าต่างที่ปรากฏอยู่ในภาพ หรือเลือก (สั่งให้) หมุนตัวหรือเปลี่ยนทิศทาง การกระทำเหล่านี้จะทำให้ภาพเปลี่ยนไป เช่น กลายเป็นภาพในห้องอีกห้องหนึ่ง หรือภาพการเดินทาง จุดมุ่งหมายของเกมชนิดนี้คือ ฟันฝ่าอุปสรรคต่าง ๆ จนกว่าจะพบกับความสำเร็จ เกมประเภทนี้ซับซ้อนมากขึ้นเมื่อมีตัวแสดงหลายตัว คอมพิวเตอร์จะเลือกการกระทำของตัวแสดงอื่น ๆ ที่เหลืออยู่ ผลของการกระทำของตัวแสดงตัวอื่น ๆ นี้จะทำให้สภาวะเปลี่ยนแปลงไป ผู้เล่นต้องรู้อุปนิสัยและคุณสมบัติของตัวแสดงต่าง ๆ ทั้งนี้เพื่อคาดเดาการกระทำของตัวละครเหล่านี้ เกมชนิดนี้มีตรรกที่ซับซ้อน และต้องใช้ภาพมาแสดงมากมาย ภาพเหล่านี้ต้องสร้างไว้ล่วงหน้า และเก็บไว้ในจานแม่เหล็ก โปรแกรมจะเลือกนำมาแสดงตามสภาวะหรือเหตุการณ์ที่ผ่านไป เช่น ถ้าผู้เล่นสั่งให้หยิบปืนขึ้นมา ภาพปืนก็จะหายไป (และอาจนำมาแสดงอยู่ขอบจอด้านซ้ายเพื่อบอกว่าสิ่งนี้อยู่ในมือผู้เล่น) เวลาผู้เล่นเลือกที่จะเดินกลับเข้ามาในห้องนี้อีก ภาพปืนก็จะไม่มีแล้ว เกมชนิดนี้จึงใช้เวลาพัฒนายาวนาน

5. พัซเซิลเกม นี่เป็นเกมที่คล้ายบอร์ดเกม แต่มีผู้เล่นคนเดียวคือท่าน ภาพบนจอจะเริ่มต้นด้วยภาพที่ถูกต้อง ครบถ้วน สวยงาม ภาพจะถูกตัดออกเป็นชิ้น ๆ เหมือน JigSaw โดยทั่ว ๆ ไปจะเป็นชิ้นสี่เหลี่ยมจตุรัสขนาดประมาณ 1 นิ้ว คูณ 1 นิ้ว โปรแกรมจะให้เวลาในการดูภาพและจำไว้ เมื่อหมดเวลาแล้วโปรแกรมจะเลื่อนภาพชิ้นเล็ก ๆ เหล่านั้นสลับกันไปมา และจะปล่อยให้มีช่องว่างไว้ 1 ช่อง ผู้เล่นจะต้องสั่งเลื่อนภาพชิ้นเล็ก ๆ เหล่านี้ไปมาเพื่อทำให้มันกลับมาเรียงต่อกันจนเป็นภาพเหมือนตอนเริ่มต้น



ครับ ในเรื่องของการเลือกชนิดของเกมเราก็คงต้องใช้กฎเกณฑ์ว่า **ทำพวกที่ง่าย ๆ เสียก่อน แล้วค่อย ๆ เพิ่มความยากขึ้นไปเรื่อย ๆ เมื่อชำนาญในชนิดหนึ่งแล้วก็ขยับไปยังประเภทที่ยากขึ้น** ท่านอาจเพิ่มไปจนถึงชนิดที่ท่านชื่นชอบ แล้วก็พัฒนาตนเองอยู่ที่กลุ่มนั้น ท่านสามารถยึดเป็นอาชีพได้ หลังจากที่มีความชำนาญในระบบดอสแล้ว ท่านก็อาจเปลี่ยนไปพัฒนาบนวินโดวส์ ความชำนาญในระบบดอสเป็นข้อกำหนดที่สำคัญก่อนที่จะกระโดดไปสู่วินโดวส์ แต่บางท่านอาจสนใจที่จะเรียนรู้ให้หมดทุกประเภทของเกม

ในเรื่องของความยากง่ายนั้นเขามักจะจัดกันตามลำดับดังนี้คือ อาร์เคดเกม บอร์ดเกม พัซเซิลเกม โรลเพลอิงเกม แล้วจึงถึงซีมูเลชันเกม ท่านควรสังเกตว่าบริษัทไมโครซอฟท์นั้นใช้โปรแกรมเมอร์ฝีมือดีที่สุดอยู่ที่การพัฒนาซีมูเลชันเกม และเป็นแผนกที่ทำกำไรสูงมาก

ครับ หลังจากเลือกชนิดของเกมแล้วก็จะมีคำถามว่า **จะออกแบบอย่างไรจึงจะทำให้เกมนั้นเร้าใจผู้เล่นมากที่สุด** นี่คือหัวใจของการสร้างเกมคอมพิวเตอร์ ผมอยากจะขอพูดถึงหัวใจของเกมประเภทอาร์เคดเกมเสียหน่อย

อาร์เคดเกมนั้นต้องการ แสง-สี-เสียง-ความเร็ว-และความคิด ในอัตรา 4-2-4-4-2 คำว่า **แสง** นั้นจะหมายถึง การกะพริบหรือเปลี่ยนแปลงภาพอย่างรวดเร็วในบางจุด เช่น เมื่อผู้เล่นกดแป้นสั่งยิง เราก็จะต้องทำเป็นลำแสงที่วิ่งออกจากกระบอกปืนไปสู่เป้า มันจะต้องรวดเร็วและเห็นได้อย่างสะดุดตา แล้วก็อาจหายไปในเวลาอันรวดเร็ว สิ่งนี้ทำได้โดยการลากเส้นสีขาวหรือสีที่มีคอนทราสต์กับสีพื้นมากที่สุดเป็นช่วงสั้น ๆ ทำติดต่อกันจากปากกระบอกปืนไปสู่เป้า เราอาจลากเส้นที่ 1 ลากเส้นที่ 2 แล้วจึงหันมาลบเส้นที่ 1 แล้วจึงลากเส้นที่ 3 แล้วลบเส้นที่ 2.... ฯลฯ นี่จะทำให้เกิดเป็นภาพที่เคลื่อนที่ได้ และรวดเร็ว อาร์เคดเกมต้องการความตื่นเต้นจากแสงมากที่สุด น้ำหนักมันจึงเป็น 4 (ให้คะแนนแบบเกรดการศึกษา)

คำว่า **สี** นั้นจะหมายถึงสีและภาพบนจอภาพ อาร์เคดเกมนี้ต้องการสีฉูดฉาด แต่ไม่จำเป็นต้องเหมือนจริง น้ำหนักมันจึงเหลือแค่ 2 โดยทั่ว ๆ ไปแล้วเราต้องออกแบบให้มีภาพพื้นให้เต็มหน้าจอ แต่ไม่เคลื่อนไหว ส่วนที่เคลื่อนไหวให้มีขนาดเล็กที่สุด ทั้งนี้เพราะการทำให้ภาพเคลื่อนไหวก็คือการเขียนภาพทับลงไปโดยมีรายละเอียดที่แตกต่างกันไปทีละน้อย ๆ การเขียนภาพทับลงไปนี้จะใช้เวลามากเมื่อเป็นพื้นที่ขนาดใหญ่ การเขียนภาพแต่ละครั้งต้องเป็นเนื้อที่สี่เหลี่ยมผืนผ้าตามแนวตั้งหรือแนวนอน (จริง ๆ แล้วแนวตั้งหรือแนวนอนก็เหมือนกัน คือมันกำหนดได้ด้วยจุด 2 จุดที่อยู่ทะแยงมุมกัน) เราจะเขียนภาพเป็นบริเวณโค้งไปโค้งมาไม่ได้ การที่จะให้เห็นเป็นภาพเคลื่อนไหวอย่างนุ่มนวลต้องเขียนทับในอัตรา 10-20 ครั้งต่อ 1 วินาที ภาพขนาดใหญ่จะทำเร็วขนาดนี้ไม่ได้

คำว่า **เสียง** นั้นเราหมายถึงการทำเสียงเป็นทำนองต่าง ๆ กัน ในบางเรื่องอาจทำเป็นเสียงคำพูดเลยก็ได้ แต่เสียงนี้เราต้องการส่งผ่านข้อมูลแค่บอกว่าผู้เล่นทำถูกหรือทำผิดเท่านั้น อย่าพยามยามใส่ข้อความเป็นคำพูดที่มีความหมายซับซ้อน ทั้งนี้เพราะผู้เล่นไม่สามารถโต้ตอบกลับมาด้วยคำพูด เสียงจึงเป็นการกระตุ้นทางจิตใต้สำนึกมากกว่าเสียงที่ดังไพเราะ เช่น กรุ๊งกริ๊ง หนิงหน่อง จะบอกว่าทำถูก ทำดี เสียงที่ขัดหู เช่น บีบ-บีบ บรือ...บรือ... จะบอกว่าทำผิดหรือ

ทำไม่ดี การโปรแกรมเสียงนี้เป็นเรื่องที่ยากพอสมควร ต้องใช้วิธีลอง-ผิด-ลอง-ถูก (trial and error) มากหน่อย อาร์เคดเกมต้องการการกระตุ้นเสียงค่อนข้างสูง มันจึงมีน้ำหนักเป็น 4

คำว่า **ความเร็ว** หมายถึงความเร็วในการโต้ตอบกัน อาร์เคดเกมจะเป็นประเภทท้าทายความสามารถในการมองดูภาพ แล้วรีบตีความและรีบกดปุ่มเพื่อเอาชนะเครื่อง ส่วนใหญ่ก็เป็นเรื่องของการยิงเป้าที่วิ่งไปมา หรือกระโดดขึ้นไปบนคานหรือบันไดที่กำลังหมุนหรือเคลื่อนที่อยู่ บางคนบอกว่าเกมชนิดนี้ใครมือไวก็ได้คะแนนมากกว่า แต่จริง ๆ แล้วมันก็ไม่เสมอไป เกมทุกเกมต้องใช้แฟกเตอร์ที่ 5 ด้วยเสมอ คือต้องคิดและวางแผน อาร์เคดเกมนี้ต้องการความเร็วในการโต้ตอบระดับ 4

คำว่า **ความคิด** นั้นเราหมายถึงการบังคับให้ผู้เล่นต้องใช้ความคิด ผู้เล่นต้องสังเกตและต้องวางแผน คนเล่นเกมที่ไม่คิดและไม่วางแผนจะได้คะแนนไม่สูงสุด ผมชอบเกมประเภทที่ต้องใช้ความคิด ต้องวางแผน อาร์เคดเกมส่วนใหญ่ในตลาดจะใส่แฟกเตอร์นี้น้อย แต่ก็มีเพิ่มขึ้นเรื่อย ๆ ผมคิดว่าแฟกเตอร์นี้ควรมีน้ำหนักไม่ต่ำกว่า 2

โอเค ที่กล่าวมาทั้งหมดนั้นเป็นหลักเกณฑ์และขั้นตอนที่จะพัฒนาตนเองไปสู่การเป็นผู้เขียนโปรแกรมเกม แต่มันก็จะมีคำถามว่า เราจะคิดเกมขึ้นมาได้อย่างไร? และเมื่อคิดได้แล้วจะเลือกใช้คำสั่ง หรือวิธีการอย่างไร จึงจะทำให้เกิดผลเป็น แสง-สี-เสียง-ความเร็ว-และความคิด ได้ตามที่ต้องการ?

สองเรื่องผมจะพูดย่อ ๆ ให้ฟังเสียก่อน แต่จะให้เกิดทักษะหรือความชำนาญนั้นต้องใช้เวลา ผมมีโปรแกรมที่พัฒนาเองอยู่ประมาณ 100 เกม ในเกมเหล่านี้จะมีตัวอย่างของวิธีการวาดภาพ วิธีการทำเสียง วิธีการกระตุ้นสายตาด้วยแสง...ฯลฯ มันต้องดูจากตัวอย่างเหล่านี้ ในตอนนี้เรามาพูดหลักการกว้าง ๆ เสียหน่อย

เกมคือการท้าทาย เช่นท้าให้เอาปืนอัดลม (ใช้ลูกเป็นไม้ก๊อก) มายิงตุ๊กตาที่เคลื่อนที่บนสายพาน ท้าให้เอาเกือกม้ามาโยนคล้องเสาหลักไม้ ท้าให้เอาเหรียญมาโยนใส่หลุม

ท้าให้วิ่งไล่จับ... ฯลฯ ดังนั้น **การจะคิดหาเกมใหม่ ๆ ก็ต้องหาเรื่องประเภท "ท้าทาย" กัน** คนเรามีเรื่องที่ท้าทายกันอย่างสนุก ๆ อยู่เยอะแยะ การท้าทายนี้อาจแยกได้เป็น 2 รูปแบบ คือ **แบบเล่นสู้กัน** นั่นต้องสร้างกฎเกณฑ์ในการต่อสู้ที่เสมอหน้ากัน แต่เล่นสลับกัน คือท่านเล่น 1 ครั้ง คอมพิวเตอร์เล่น 1 ครั้ง แล้วก็สลับกันไปเรื่อย ๆ พวกนี้จะเป็นบอร์ดเกมเป็นส่วนใหญ่ การท้าทายแบบที่ 2 คือ **ฝ่ายหนึ่งเป็นเป้าอีกฝ่ายหนึ่งเป็นผู้ยิงเป้า** เกมแบบที่ 2 นี้เป็นประเภทอาร์เคดและอื่น ๆ คอมพิวเตอร์จะเป็นผู้ควบคุมเป้า ผู้เล่นจะเป็นผู้ยิงเป้า คำว่า**ยิงเป้า** ในที่นี้ไม่ได้หมายความแค่การเอาปืนหรือธนูมายิง มันมีความหมายถึงการตั้งเป้าหมายอะไรก็ได้ เช่น การที่จะเป็นเศรษฐีมีเงินล้าน การที่จะดั้นด้นไปจนถึงจุดหมายปลายทาง การที่จะอยู่รอดในสภาวะอันมีภัยรอบด้านในเวลาอันใดอันหนึ่ง เมื่อมีเป้าหมายแล้วก็มากำหนดว่าผู้เล่นจะทำอะไรได้บ้าง แล้วก็มาสร้างภาพบนจอภาพเพื่อบอกสถานการณ์ แล้วผู้เล่นก็กดปุ่มหรือแป้นต่าง ๆ เพื่อให้เกิดผลตามที่ต้องการ การตอบแทนผลของการกระทำนั้นจะขึ้นอยู่กับเวลาและความถูกต้องของการกดปุ่ม ถ้ากดเร็วเกินไปหรือช้าเกินไปก็จะพลาดเป้า ซึ่งจะต้องถูกหักคะแนน การกดได้ในเวลาอันพอเหมาะก็จะถูกเป้า และได้รับคะแนนเพิ่ม

ครับ ผมบอกได้เลยว่า การคิดสร้างเกมขึ้นมานั้นไม่ยากเลย และเกือบทุกครั้งเราสามารถนำมาเขียนเป็นโปรแกรมเพื่อให้คนอื่นมาทดลองฝีมือได้ แต่มันเป็นเรื่องเศร้าอยู่หน่อยหนึ่ง คือ หลังจากใช้เวลาคร่ำเคร่งอยู่ที่กระดาษร่างและหน้าจอคอมพิวเตอร์อยู่หลาย ๆ วัน เกมนั้นก็จะเสร็จสมบูรณ์ แต่มันจะไม่ใช่เกมที่ท่านจะเล่นสนุกได้มากนัก คนทำเกมนั้นเล่นเกมของตัวเองไม่สนุก มันไม่สนุกเพราะรู้ทางหนีทีไล่จนเกือบหมดแล้ว

ยิ่งไปกว่านั้น หลังจากเขียนโปรแกรมเกมไปได้มาก ๆ แล้ว ก็จะพบว่า เกมที่มีอยู่ในท้องตลาดนั้นมันดูน่าเบื่อขึ้นทุกวัน

เรื่องนี้เป็นเรื่องเศร้าจริง ๆ การเขียนตำราของผมมันจึงค้างคามานานด้วยเหตุผลนี้

สวัสดีครับ